RoboCup2004
Rescue Robot League Competition
Lisbon, Portugal
June 27 – July 5, 2004
www.robocup2004.pt

RoboCupRescue - Robot League Team
RFC Uppsala, Sweden

Marie Carlson, Carl Christiansen, Andreas Persson, Erik Jonsson, Mattias Seeman, Håkan Vindahl

Uppsala University
Department of Information Technology
Box 337, SE-751 05 Uppsala, Sweden
cach8421@student.uu.se
http://hermod.robocup.it.uu.se

Abstract. The robot system designed by RFC Uppsala for the 2004 RoboCup Rescue Real League is a fully autonomous team of four robots. The team works together to generate a map by using ultra sonic sensors.

Introduction

RFC Uppsala is a team from Uppsala University consisting of senior students attending the Master of Science program of Information Technology. Previous years RFC Uppsala has competed in the RoboCup Soccer Middle Size League, but this year the 20 weeks long project is aiming to develop a robot system designed for the RoboCup Rescue Real League. The experiences from previous years' projects are very valuable.

The goal is to develop a fully autonomous system consisting of four robots that have the ability to work as a team. The team should search the area and locate victims. Using ultra sonic sensors a map is generated to assist human rescue teams in their operation.

To develop effective search strategies a simulator where the robot software is evaluated has been constructed. This saves much time since experiments can be performed in a faster than real-time environment and parallel to the development of the hardware platform.

1. Team Members and Their Contributions

- Jakob Carlström — Advisor
- Per Halvarsson — Advisor
- Rasha Alshammari — Software developer
- Gustav Björcke — Software developer

- Marie Carlson — Hardware developer
- Carl Christiansen — Software developer
- Pär Johansson — Software developer
- Erik Jonsson — Hardware developer
- Jonas Jämtberg — Software developer
- Jonas Nilsson — Hardware developer
- Andreas Persson — Software developer
- Jon Rönnols — Software developer
- Mattias Seeman — Hardware developer
- Mathias Thore — Hardware developer
- Håkan Vindahl — Hardware developer

2. Operator Station Set-up and Break-Down (10 minutes)

The operator station consists only of a laptop with the human-robot interaction software installed and a small portable printer. Since the robots are fully autonomous the operator station is not be manned during the missions if there is no system failure forcing human operators to take manual control over the robots. The whole team of robots is controlled by, at most, one operator working on one laptop.

The robots are carried into the arena and be placed in a special formation in which the robots calibrate to a common reference system. This is the time critical part of the setup.

During initialization of robots there is a defined rank between members of the robot team. Highest ranking robot starts in the origin, and the other robots are set up in a pattern such that they know there position relative to the first robot.

3. Communications

The robots use WLAN communication with the 802.11A standard. The communication between the robots is done in infrastructure mode. The robots communicate with each other and the operator interface through a base station. This limits the range of operation slightly.

4. Control Method and Human-Robot Interface

The team is designed to be fully autonomous, however there is a human-robot interface mainly used for development and debugging. During missions no human operator is needed unless some system fails.

The human-robot interface controls the whole team of robots from one laptop over the WLAN interface. Status of all subsystems for each robot is available from the human-robot interface and advanced functions for steering the robots are available.

The operator can choose to control the robots exact paths, but also let the robots calculate their own paths and search behavior if that is preferred.

When in autonomous mode a planner runs on the robot that is selected as master at the current time, which robot that is master can change during the mission. If the master is lost another robot is automatically appointed to master. The planner allocates search areas for the robots in the team.

To effectively search the area the master allocates one section of the total area to each robot, when a slave robot has been allocated an area the local robot navigator calculates the path used to reach and search the given area.

5. Map Generation/Printing

The map is mainly generated from the ultrasonic sensors. The robot receives readings of angles and distances to the nearest obstacle for each angle from the ultrasonic sensors, the points in the map at the given distance is calculated and marked as obstacles. All points between the robot and the obstacle are marked as free from obstacles. Data from the pyroelectric sensor are used to mark potential victims.

Morphological methods for image analysis are used to improve the quality of the map and different colours are used to mark the obstacles. If the analysis shows that earlier discovered areas do not match the new data the main system is alerted of this and take actions to locate its own position to be able to correct data.

A global map consisting of the individual robots local maps are generated during the mission. Analysis and synchronization of the global map are distributed within the team to improve performance. When the server is reachable within the network analysis and synchronization can be done on the server to relieve the robots from some of the work.

The server also analyzes the map with statistical methods and image analysis after the robots have finished their mission to enhance the quality even more before it is printed from the human-robot interaction software. Related previous work includes [1], [2] and [3].

6. Sensors for Navigation and Localization

All the sensors are connected to the central computer through a CANbus that runs through the robot. The central computer receives the data and processes it.

Ultrasonic Sensors

The robots have active ultrasonic sensors mounted on the front of the robot. These return distances and angles to objects. Their purpose is to supply information necessary for mapping and navigation of the robots and also to enable them to avoid collision.

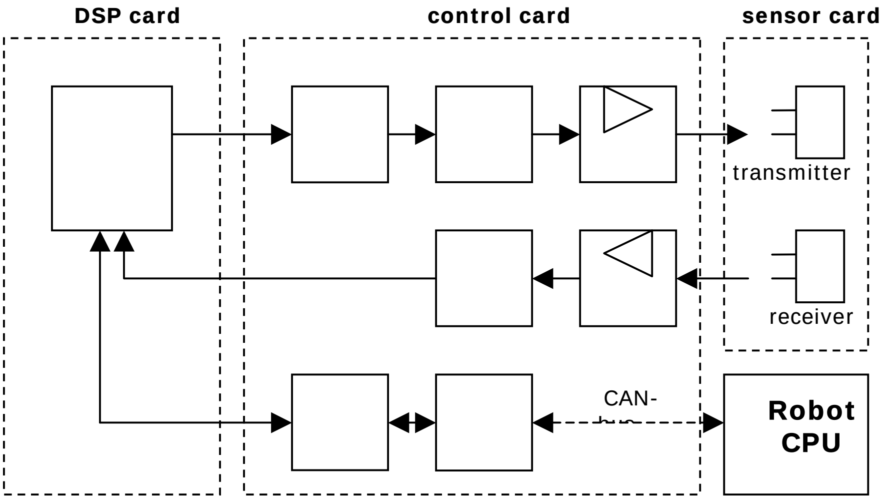

Fig. 1. The hardware for the ultrasonic sensors consists of a number of different cards.

The sensor card consists of transmitters and receivers. The transmitters emit a burst of 40 kHz sound when the control card issues a signal. The receivers listen for an echo. It is possible to steer the ultrasonic beam ±30 degrees, which means that a sector of about 60 degrees in front of the robot can be searched. The ultrasonic system can reach a distance from about half a meter up to about 4 meters.

The transmission of ultrasound is initiated by a DSP by sending a trigger signal to the control card. When an echo returns it is analyzed to detect if there were any reflections from objects. If an object is found, an angle and distance to the object is calculated by the DSP and then sent onwards to the robot's CPU via the CANbus.

The CAN card handles the communication between the DSP and the CANbus. The CAN card receives information from the DSP via UART with information about the object that is detected by the ultrasonic system and send this information out on the CANbus to the central computer.

Infrared Sensors

The purpose of the active infrared sensors is to detect holes and steep slopes so that the robots can avoid them. They are also supposed to act as a complement to the ultrasonic system which operates from half a meter. The infrared sensors operate on distances up to 80 cm. Therefore, the infrared sensors can enable collision detection and avoidance at close distances. The infrared sensors are placed in a way such that all directions around the robot are observed. Close to each tire, sensors are placed to prevent the robot from entering holes. Sensors are placed forwards, as a complement to the ultrasonic system, to detect obstacles. On the sides of the rear end of the robot sensors are placed to avoid collisions when the robot turns and also sensors facing backwards to prevent from collisions while reversing.

The sensors, which are analogue, work at distances from 10 cm to 80 cm. They report an output voltage that can be transformed into a distance. Each sensor is connected to an input channel on an AVR microcontroller. The microcontroller uses analogue to digital conversion to read the output voltage value from an analogue infrared sensor. The controller calculates the distances that the output voltage values of the sensors correspond to and send distance reports on the CANbus whenever the central computer wants information.

Motor Positioning Calculation

The electrical motors are controlled by a dedicated card, which also provides sensor data to the navigation system.

A pair of AVR microcontrollers are mounted on the circuit board controller for the motors to take care of the pulses from the encoders and to make positioning calculation. The microcontrollers communicate through an i2c bus. The encoder information is being used to instantly adjust the motor speed and calculate the position of the robot. The solution with motor controller and encoder feedback on the same circuit board gives the possibility to drive the motors very exactly, which gives a very high precision in the movements of the robot.

The CAN messages to control the movement contain three parts: type of movement, speed and angle/distance/radius. The types movements are: go forward, go backward, turn left, turn right, rotate left, rotate right and stand still. From the engine controller CAN-messages containing position information is being sent continuously. The pace of position information packets being sent is controlled by how the robot is moving. The position packets can also be sent on demand.

Position Recovery

If one of the robots in a team for any reason finds that its local coordinate system is misaligned with the team's global coordinates, it has the ability to request help to find its correct position from one the other robots.

With the the help of infra-red beacons and a pair of receivers on every robot, a lost robot can triangulate its own position in relation to another robot. Knowing the position and angle of the other robot, a correct position can be calculated. This system shares micro-controller and sensor arm with the pyro-sensor module (see Sensors for Victim Identification).

7. Sensors for Victim Identification

Pyro-sensors

Pyro-electric sensors are used to provide a way for the robots to identify and localize victims. A pair of sensors is used in stereo to provide accurate information on the position of a victim.

The pyro-sensor is a passive infrared sensor responsive to the wavelength of IR light emitted by a warm human body. Since the sensor reacts to differences in temperature between two areas, light from these two areas is focused upon the receptive areas of the sensor through a thin sliver of a fresnel lens, and a pair of sensors sweep their surroundings mounted on a stepper motor.

The stepper motor is controlled by an AVR microcontroller, which also interprets the analog signal from the sensors. The position of a victim is found by triangulation. The AVR then in turn communicates the positions of found victims to the robot's central CPU via the CANbus.

Microphone

A microphone is used to localize victims. This helps in finding hidden or entombed living humans, whose only way of telling where they are is through tapping or moaning.

Audio is continuously sampled through a single directional microphone facing forward. The audio stream is then processed for features distinguishing human voices and tapping sounds from background noises.

Camera

A regular web camera, connected by USB to robot central computer is used. The camera is used to document sites where victims are discovered to further help the rescue operation.

The camera is mounted on the front of the robot and can be triggered by software, i.e. on victim discovery by the pyro-sensor.

8. Robot Locomotion

As shown in figure 2 the robots are constructed with two front driving wheels and one rear wheel. The rear wheel follows the robot like the wheels on a shopping cart or a wheelchair. The positive effects of using just one rear wheel compared to two rear wheels is that the robot size and turnaround radius decrease.

Two motors are used, one for each driving wheel. The robot turns by rotating the driving wheels at different velocities.

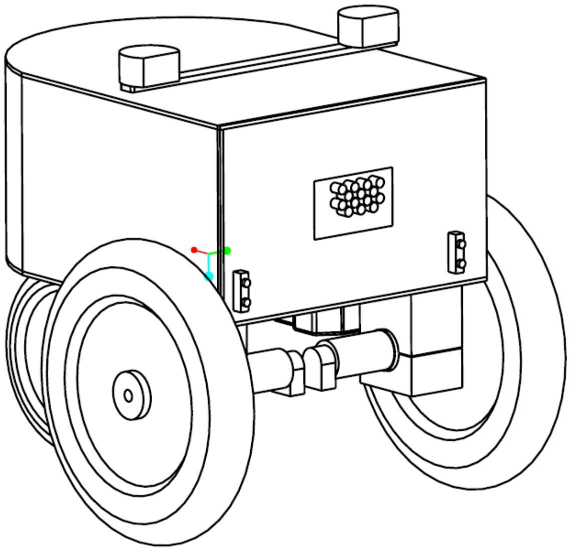

Fig. 2. The robot basic design.

9. Other Mechanisms

Motors

The motor packs consist of a motor, model RE40 from Maxon. The motor is a 150 Watt, 48 Volt DC motor. A planetary gear head is mounted on the motor. The motor is also equipped with a Maxon encoder which gives an output of 500 pulses per turn on the motor. The engines are strong enough to drive a robot weighing 20kg over obstacles up to 75mm high without problems.

The engines are controlled by a single circuit board controller. The board is equipped with a CAN interface to communicate with other nodes on the robot. A microcontroller is being used to handle the CAN communication and to control the motors.

10. Team Training for Operation (Human Factors)

Since the system is fully autonomous minimal training is required for use. To operate the system a human-robot graphical interface is used where all robots can be controlled simultaneously. The interface is easy to use, but advanced functions for accessing robot subsystems for debugging purposes demands good knowledge of the system.

The operators don't need any extra training in addition to the normal testing before the competition. Tests includes however setup time and how to handle failure on the robot subsystems that might force an operator to take control of the team. We select the person to operate the system in case of failure of the autonomous system by timed test runs.

11. Possibility for Practical Application to Real Disaster Site

Best chances for success are in buildings where ground obstacles are small. The system works perfectly without lighting since ultra sonic sensors are used for map building.

12. System Cost

The system cost per robot is approximately 2800 €. This does not include the operator station, which consists of a standard PC with an IEEE802.11a WLAN card and a printer. For further information see appendix A.

References

[1] Philip Althaus and Henrik I. Christensen “Automatic Map Acquisition for Navigation in Domestic Environments”, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1551-1556, 2003.

[2] Bing-Ran Zuo and I-Ming Chen “Predict Sonar Returns for Autonomous Robot Navigation in Unstructured Enviroments” http://155.69.254.10/users/risc/Pub/Conf/00-c-icarcv-sonar.pdf

[3] Bing-Ran Zuo and I-Ming Chen “A Dynamic Reachability Test for Sensor-Based Navigation with Unknown Obstacle Boundaries”, in Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 2030-2035 vol 4, 2001

Appendix A

Number of items	Module	price/part exkl.moms (SEK)	tot price exkl. moms (SEK)	tot price (EUR)
	Wlancard			
1	Orinoco Combocard Gold 802.11a/b Cardbus	650,00	**650,00**	**71,50**
	Webcam/mic			
1	Philips ToUcam Pro PCVC 740k	621,00	**621,00**	**68,31**
	Ultrasonic			
	Sensorcard:			
14	Polaroid 40LT10 part no. 626997	68,00	952,00	104,72
2	Receivers	56,00	112,00	12,32
	Tot Sensorcard		1 064,00	117,04
	Controllcard:			
1	ADC	108,00	108,00	11,88
2	Transistor	13,80	27,60	3,04
1	Op Amp	10,20	10,20	1,12
1	Op Amp	5,32	5,32	0,59
1	AVR	55,80	55,80	6,14
2	Plug 4/4	14,60	29,20	3,21
4	Shift register	32,40	129,60	14,26
1	DC/DC converter	350,00	350,00	38,50
1	Oscillator	74,30	74,30	8,17
1	AVR CAN	151,00	151,00	16,61

1	CAN interface	33,40	33,40	3,67
1	Circuit board	200,00	200,00	22,00
	Tot Controllcard		1 174,42	129,19
	CAN-card:			
1	CAN controller	20,00	20,00	2,20
2	Optocoupler	44,80	89,60	9,86
1	CAN transceiver	33,40	33,40	3,67
1	Reset circuit	24,80	24,80	2,73
1	AVR Microcontroller	165,00	165,00	18,15
1	RS-232 sender/receiver	23,30	23,30	2,56
1	Circuit board	100,00	100,00	
	Tot CAN-card		456,10	50,17
	DSP-card:			
1	ADSP-21065L SHARC, EZ-LAB	2 500,00	2 500,00	275,00
	Tot Ultrasonic		**5 194,52**	**571,40**
	IR			
8	SHARP IR sensor	44,25	**354,00**	**38,94**
	IR robot recognition			
7	IrED OP290A	14,30	100,10	11,01
2	Phototransistor OP598A	17,30	34,60	3,81
2	Reflector for 5 mm LED	5,22	10,44	1,15
1	Binocular for parts	69,00	69,00	7,59
1	Effecttransistor, Darlington BD677A	5,82	5,82	0,64

	Tot IR robot recognition		**219,96**	**24,20**
	Tires			
2	Front tires	142,50	285,00	31,35
1	Rear tire	273,75	273,75	30,11
	Tot Tires		**558,75**	**61,46**
	Motors (Stork)			
2	DC-motor	1 200,00	2 400,00	264,00
2	Planetary Gear	1 065,00	2 130,00	234,30
2	Puls sensor	410,00	820,00	90,20
2	Assembly set	36,00	72,00	7,92
	Tot motors		**5 422,00**	**596,42**
	Motorcontroller			
1	Parts	2 500,00	2 500,00	275,00
1	PCB	875,00	875,00	96,25
	Tot Motorcontroller		**3 375,00**	**371,25**
	The frame			
1	Work + material	7 500,00	**7 500,00**	**825,00**
	Batteries			
2	Batteries	575,00	1 150,00	126,50
1	Charger	225,00	225,00	24,75
2	Holds	25,00	50,00	5,50

	Tot Batteries		**1 425,00**	**156,75**
	Pyro sensor			
2	Operational amplifier	5,82	11,64	1,28
2	Rail-to-rail op	7,72	15,44	1,70
2	Pyroelectric IR-sensor (Nippon Ceramic)	49,90	99,80	10,98
2	Fresnel lens (Nippon Ceramic)	45,60	91,20	10,03
2	Stepper motor driver (Allegro)	45,30	90,60	9,97
1	Stepper motor	512,00	512,00	56,32
	Tot Pyro		**820,68**	**90,27**
	RFC-CAN cards			
3	Cards	97,50	292,50	32,18
3	AVR Microcontroller	185,00	555,00	61,05
3	CAN controller	20,00	60,00	6,60
3	CAN transceiver	33,40	100,20	11,02
3	Reset circuit	24,80	74,40	8,18
6	Optocoupler	44,80	268,80	29,57
	Tot RFC-CAN cards		**1 350,90**	**148,60**
	CPU (sponsored prices)			
1	32MB SODIMM memory expantion	179,26	179,26	19,72
1	Hectronic H6015 central computer	2 014,00	2 014,00	221,54
1	CompactFlash 128MB	409,00	409,00	44,99
1	Hectronic H7006 CAN-card PC/104	350,00	350,00	38,50
1	Cables	300,00	300,00	33,00

1	Assembly – testing	150,00	150,00	16,50
	Tot CPU		**3 402,26**	**374,25**
	TOTAL SYSTEM COST		**30 019,07**	**3 302,10**